

IDENTIFIKASI POLA DISTRIBUSI CURAH HUJAN MAKSIMUM DAN PENDUGAAN PARAMETERNYA MENGGUNAKAN METODE *BAYESIAN MARKOV CHAIN MONTE CARLO*

Moch. Abdul Mukid¹, Yuciana Wilandari¹

¹Staf Pengajar Jurusan Statistika UNDIP
mamukid@yahoo.com

Abstract

A good understanding of the pattern and the distribution of maximum rainfall is an important thing, especially for the management of regional water resources. In this study, we not only identify the distribution of maximum rainfall, but also estimate the parameter of its distribution. The research was conducted in the Grobogan District. Maximum rainfall in the district of Grobogan from 2006 to July 2012 was very varied, but over the years have a pattern unlikely to change. Highest maximum rainfall ranged in December, January, February and March while the lowest rainfall maskimum normally be in June, July and August. By using the Kolmogorov-Smirnov test on the significance level of 5% is known that the maximum rainfall from 2006 to 2012 in the District Grobogan follow a normal distribution with a value of D statistics is 0.089. This statistic produces a significance value of 0.518. By using the Bayesian Markov Chain Monte Carlo obtained the value for the parameter mean of normal distribution is 46.269 mm with a standard error reach into 4.005 mm.

Key Words: Maximum Rainfall, Bayesian Markov Chain Monte Carlo, Metropolis-Hastings Algorithm

1. Pendahuluan

Fenomena hidrologi adalah fenomena yang sangat rumit dan tidak akan pernah sepenuhnya bisa dimengerti. Daur hidrologi dapat disederhanakan sebagai suatu sistem, yang komponen-komponennya berupa curah hujan, penguapan, aliran dan tahapan-tahapan lain dari daur hidrologi. Curah hujan adalah banyaknya air yang jatuh ke permukaan bumi. Derajat curah hujan dinyatakan dengan jumlah curah hujan dalam suatu satuan waktu. Biasanya satuan yang digunakan adalah mm/jam. Dalam meteorologi butiran hujan dengan diameter lebih dari 0.5 mm disebut hujan dan diameter antara 0.05 – 0.1 mm disebut gerimis. Semakin besar ukuran butiran hujan maka semakin besar pula kecepatan jatuhnya^[7].

Pemahaman yang baik terhadap pola dan distribusi curah hujan maksimum adalah suatu hal yang penting terutama untuk manajemen sumber daya air dari suatu wilayah. Pengetahuan terhadap karakteristik curah hujan ini, baik secara temporal maupun spatial memainkan peran penting dalam hal perancangan dan pengoperasian sistem pertanian, telekomunikasi maupun sistem kendali kualitas air. Dalam hal ini curah hujan diasumsikan bersifat stokastik^[6].

Dalam penelitian ini selain melakukan identifikasi jenis distribusi dari data curah hujan maksimum, juga akan dilakukan pendugaan parameter dari distribusinya tersebut. Parameter adalah besaran yang mencirikan karakteristik dari sebuah populasi^[9]. Nilai parameter yang sesungguhnya biasanya tidak diketahui. Oleh karena itu nilai parameter tersebut perlu diduga.

Metode pendugaan parameter populasi yang sering digunakan adalah metode *maximum likelihood*^[1]. Metode ini didasarkan pada pendekatan statistika klasik dimana pendugaan parameter dan inferensinya berdasarkan informasi yang termuat dari sampel yang diperoleh dan mengabaikan informasi awal (*prior*) dari peneliti maupun petugas lapangan. Menurut Casella dan Berger^[3] pendekatan klasik ini memiliki kelemahan dalam hal interpretasi terhadap selang kepercayaan dari parameter distribusi. Oleh karena itu, dalam penelitian ini pendugaan parameter distribusi dilakukan dengan metode Bayes dan *Markov Chain Monte Carlo* (MCMC).

Kelebihan metode Bayes terletak pada kemampuannya dalam mengakomodasi informasi awal yang dimiliki oleh peneliti. Kesimpulan akhir dari nilai dugaan parameter distribusi didasarkan tidak hanya pada informasi dari sampel saja tetapi juga didasarkan atas informasi subyektif dari peneliti. Informasi awal ini (*prior information*) biasanya dikuantifikasi dalam bentuk distribusi dari parameternya^[2].

Dalam kondisi dimana distribusi prior dari parameter adalah sekawan (*conjugate*) dari distribusi sampel acaknya (*likelihood*) maka komputasi dari distribusi posterior parameter relatif tidak menemui kesulitan yang berarti, karena distribusi posteriornya akan berada pada kelas yang sama dengan distribusi priornya. Namun jika distribusi prior dari parameter bukan merupakan sekawan dari distribusi sampel acaknya maka komputasi dari distribusi posterior parameternya akan menemui kesulitan. Kondisi ini makin dipersulit lagi jika melibatkan banyak parameter. Dalam situasi inilah simulasi *Markov Chain Monte Carlo* (MCMC) diperlukan.

2. Tinjauan Pustaka

2.1. Algoritma Metropolis-Hastings (M-H)

Pada awalnya Metropolis dkk. pada tahun 1950 memformulasikan algoritma *Metropolis* dengan mengenalkan rantai Markov berdasarkan metode-metode simulasi yang digunakan dalam ilmu pengetahuan alam. Kemudian Hastings pada tahun 1970 menggeneralisasi metode tersebut yang terkenal dengan nama algoritma *Metropolis-Hastings*. Selanjutnya pada tahun 1995, Green menggeneralisasi algoritma *Metropolis-Hastings* dengan mengenalkan algoritma *Metropolis-Hastings reversible jump* untuk sampling dari ruang parameter dengan dimensi yang berbeda^[5].

Misalkan fungsi densitas dari distribusi targetnya adalah $f(\theta|\mathbf{x})$ dimana sampel yang diinginkan akan dibangkitkan dengan ukuran T dan misalkan pula $\theta^{(t)}$ adalah vektor dari nilai-nilai yang dibangkitkan di iterasi ke- t dari algoritma. Algoritma *Metropolis-Hastings* dapat dijelaskan dengan mengikuti langkah-langkah iterasi sebagai berikut:

1. Ambil nilai awal $\theta^{(0)}$
2. Untuk $t = 1, \dots, T$, ikuti dan ulangi langkah-langkah:
 - a. Mengatur $\theta = \theta^{(t-1)}$.
 - b. Bangkitkan nilai calon parameter θ' dari distribusi proposal $q(\theta'|\theta)$.
 - c. Hitung $\alpha = \min\left(1, \frac{f(\theta'|\mathbf{x})q(\theta|\theta')}{f(\theta|\mathbf{x})q(\theta'|\theta)}\right)$.
 - d. Bangkitkan u dari distribusi Seragam(0,1)
 - e. Jika $u \leq \alpha$, maka $\theta^{(t)} = \theta'$ dan jika tidak maka $\theta^{(t)} = \theta^{(t-1)}$.

Karakteristik penting dari algoritma ini adalah bahwa tidak perlu mengevaluasi konstanta normalisasi $f(\theta|\mathbf{x})$, karena akan dihapus di α . Iterasi dari Algoritma *Metropolis-Hastings* akan konvergen ke distribusi target tanpa memperhatikan apapun distribusi proposal q yang dipilih. Namun dalam prakteknya, pemilihan distribusi proposal penting karena pemilihan yang salah akan memperlambat konvergensi terhadap distribusi target.

2.2. Pemilihan *Proposal Distribution*

Untuk mengimplementasikan algoritma *Metropolis-Hastings* perlu ditentukan distribusi proposal yang tepat. Distribusi proposal (q) adalah sebuah distribusi yang digunakan untuk membangkitkan calon anggota sampel. Beberapa metode yang sering digunakan dalam pemilihan distribusi proposal untuk algoritma *Metropolis-Hastings*, yaitu rantai *random-walk*, *independence sampler*, metode *acceptance-rejection*, rantai autoregresif dan lain-lain.

2.2.1 *Random-walk Metropolis*

Dalam algoritma Metropolis yang asli, hanya distribusi proposal yang simetris dengan tipe $q(\theta'|\theta) = q(\theta|\theta')$ yang diperhatikan. *Random-walk Metropolis* merupakan kasus khusus dengan $q(\theta'|\theta) = q(\theta - \theta'|\theta)$. Oleh karena itu hanya menghasilkan probabilitas penerimaan yang hanya tergantung pada distribusi target, yaitu:

$$\alpha = \min\left(1, \frac{f(\theta'|\mathbf{x})}{f(\theta|\mathbf{x})}\right) \quad (1)$$

Biasanya distribusi proposal ini adalah normal multivariat $q(\theta'|\theta) = N_d(\theta, \mathbf{S}_\theta)$ dengan d adalah dimensi dari θ . Matriks kovarian \mathbf{S}_θ mengontrol kecepatan konvergensi algoritma. Nilai varian dari komponen θ menentukan seberapa dekat nilai yang diusulkan dari nilai sekarang. Nilai varian yang kecil akan menghasilkan rating penerimaan yang tinggi tetapi konvergensinya lambat karena algoritma akan memerlukan jumlah iterasi yang besar untuk mengeksplor ruang parameter. Pada kasus ini, autokorelasi yang besar akan muncul dalam analisis output. Dengan kata lain, nilai varian distribusi proposal yang besar pada varian tertentu dari θ akan menghasilkan rating penerimaan yang rendah.

2.2.2. *Independence Sampler*

Independence sampler adalah sebuah algoritma *Metropolis-Hastings* dimana distribusi proposalnya tidak bergantung pada *state* rantai sebelumnya $\theta^{(t-1)}$. Sebagai contoh pilihan yang sering dilakukan adalah distribusi normal multivariat dengan tipe $\theta' \sim N_d(\bar{\theta}, S_{\bar{\theta}})$. Parameter (mean dan varian) distribusi proposal ini dapat diperoleh menggunakan metode pendekatan atau percobaan sebelumnya yang tersedia atau informasi dari ahli.

Independence sampler efektif ketika distribusi proposal $q(\theta)$ pendekatannya bagus terhadap distribusi target $f(\theta|\mathbf{x})$. Densitas proposal independen yang bagus dapat didasarkan atas pendekatan Laplace. Pada umumnya proposal yang sukses dapat diperoleh dari distribusi normal multivariat dengan mean sama dengan modus target $\tilde{\theta}$ dan matriks presisinya adalah:

$$\mathbf{H}(\tilde{\boldsymbol{\theta}}) = \left(- \frac{\partial^2 \log f(\boldsymbol{\theta}|\mathbf{x})}{\partial \theta_i \partial \theta_j} \right)_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}} \quad (2)$$

Konsekuensinya, sebuah proposal yang efisien diberikan oleh:

$$q(\boldsymbol{\theta}) = N_d \left(\tilde{\boldsymbol{\theta}}, [\mathbf{H}(\tilde{\boldsymbol{\theta}})]^{-1} \right) \quad (3)$$

Peluang penerimaan, ketika transisi yang diinginkan dari \mathbf{x} menjadi \mathbf{x}' diberikan oleh:

$$\alpha = \min \left(1, \frac{f(\boldsymbol{\theta}'|\mathbf{x})q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{f(\boldsymbol{\theta}|\mathbf{x})q(\boldsymbol{\theta}'|\boldsymbol{\theta})} \right) \quad (4)$$

Yang dapat dituliskan kembali sebagai:

$$\alpha = \min \left(1, \frac{\omega(\boldsymbol{\theta}')}{\omega(\boldsymbol{\theta})} \right) \quad (5)$$

dengan $\omega(\boldsymbol{\theta}) = f(\boldsymbol{\theta}|\mathbf{x})/q(\boldsymbol{\theta})$ merupakan rasio antara distribusi target dengan distribusi proposal dan bobotnya equivalen digunakan sampel yang penting.

2.3. Analisis Output

Dengan menjalankan algoritma MCMC, nilai-nilai simulasi $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(t')}$ masing-masing terdistribusi secara kira-kira ke distribusi target $f(\boldsymbol{\theta}|\mathbf{x})$. Untuk algoritma MCMC yang digunakan pada pembahasan ini, diketahui bahwa distribusi dari nilai simulasi pada iterasi ke- t , $\boldsymbol{\theta}^{(t)}$, akan konvergen ke pengambilan dari distribusi target jika t mendekati tak hingga. Sayangnya, hasil teori ini tidak memberikan petunjuk yang mudah digunakan pada bagaimana menentukan jika simulasi sederhana memberikan sebuah perkiraan yang beralasan untuk densitas $f(\boldsymbol{\theta}|\mathbf{x})$. Hal ini berarti tidak dapat diketahui seberapa panjang rantai harus berjalan. Berikut ini penjelasan secara singkat beberapa masalah yang penting pada interpretasi output MCMC serta penjelasan grafik dan diagnostik numerik untuk menduga konvergensi.

2.3.1. Cek Konvergensi

Istilah konvergensi algoritma mengacu pada apakah algoritma telah mencapai distribusi targetnya. Jika benar, maka pembangkitan sampel diperoleh dari distribusi target yang benar. Oleh karena itu, memantau konvergensi dari algoritma adalah penting untuk menghasilkan sampel dari distribusi target yang dicari.

Ada beberapa cara untuk memantau konvergensi algoritma. Metode yang sering digunakan, yaitu:

1. Trace Plot

Nilai-nilai simulasi dari $\boldsymbol{\theta}$ yang diperoleh pada permulaan MCMC tidak berasal dari distribusi target. Namun, setelah sejumlah iterasi dilakukan (*the burn-in period*), pengaruh dari nilai awal berkurang dan distribusi iterasi yang baru mendekati distribusi target yang benar. Salah satu cara pendugaan *burn-in period* adalah dengan memeriksa *trace plot* nilai simulasi dari komponen (atau beberapa fungsi lainnya) $\boldsymbol{\theta}$ terhadap jumlah iterasi. *Trace plot* merupakan sebuah plot dari iterasi versus nilai yang telah dibangkitkan. *Trace plot* menjadi penting, terutama ketika algoritma MCMC dimulai dengan nilai-nilai parameter yang jauh dari pusat distribusi target. Pada kasus seperti itu, nilai-nilai simulasi dari $\boldsymbol{\theta}$ pada awal iterasi algoritma akan menyimpang dari daerah ruang parameter dimana distribusi

target dipusatkan. Sebuah tren naik atau turun pada nilai parameter pada *trace plot* menunjukkan bahwa *burn-in period* belum selesai. Jika tren-tren seperti ini muncul, maka penting untuk menghilangkan bagian awal dari rantai, karena nilai-nilai awal ini tidak menunjukkan perkiraan sampel yang benar dari distribusi target. Dengan kata lain, jika semua nilai-nilai berada dalam sebuah daerah tanpa keperiodikan yang kuat dan (secara khusus) kecenderungan, maka dapat diasumsikan telah tercapai konvergensi.

2. Autokorelasi

Perhatian yang kedua pada analisis output algoritma MCMC adalah tingkat autokorelasi pada nilai-nilai sampel. Pada algoritma *Metropolis-Hastings*, nilai simulasi θ pada iterasi ke- $(t+1)$ bergantung pada nilai simulasi pada iterasi ke- t . Jika pada rantai terdapat korelasi yang kuat di antara nilai-nilai yang beruntun, maka kedua nilai beruntun tersebut memberikan informasi hanya secara marginal mengenai distribusi target dan bukan nilai dari sebuah simulasi tunggal. Korelasi yang kuat di antara iterasi yang beruntun menunjukkan bahwa algoritma masih berada pada daerah tertentu dari ruang parameter dan mungkin membutuhkan waktu yang lama untuk penyampelan dari keseluruhan daerah distribusi.

Statistik yang umum digunakan untuk mengukur tingkat ketergantungan diantara pengambilan berurutan pada rantai adalah autokorelasi. Autokorelasi mengukur korelasi diantara kumpulan nilai-nilai simulasi $\{\theta_j^{(t)}\}$ dan $\{\theta_j^{(t+L)}\}$, dimana L merupakan ukuran lag dari iterasi terpisah pada dua kumpulan nilai-nilai. Untuk komponen tertentu, fungsi autokorelasi dapat dihitung sebagai fungsi nilai-nilai yang berbeda dari lag, L . Untuk komponen j , autokorelasi L dapat diduga dengan:

$$r_{jL} = \frac{T' - L}{T'} \frac{\sum_{j=1}^{T'-L} (\theta_j - \bar{\theta})(\theta_{j+L} - \bar{\theta})}{\sum_{j=1}^{T'-L} (\theta_j - \bar{\theta})^2} \quad (6)$$

dengan $\bar{\theta}$ merupakan rata-rata dari nilai-nilai simulasi. Nilai autokorelasi untuk lag 1 akan hampir selalu menjadi positif. Namun, jika rantai yang digunakan cukup panjang, nilai-nilai autokorelasi akan berkurang menuju nol selama nilai lag ditingkatkan.

3. Ergodic Mean Plot

Istilah *ergodic mean* menunjukkan nilai rata-rata sampai dengan *current iteration*. *Ergodic mean plot* adalah plot antara nilai rata-rata di atas dengan iterasinya. Jika *ergodic mean* stabil setelah beberapa iterasi, maka ini merupakan sebuah indikasi konvergensi dari algoritma.

2.3.2. Pendugaan Parameter

Dengan menjalankan sebuah algoritma MCMC, nilai-nilai simulasi $\theta^{(1)}, \dots, \theta^{(T')}$ masing-masing terdistribusi secara kira-kira ke distribusi posterior $f(\theta|\mathbf{x})$. Penduga untuk komponen ke- j dari vektor parameter θ adalah

$$\bar{\theta}_j = \frac{\sum_{t=1}^{T'} \theta_j^{(t)}}{T'} \quad (7)$$

dengan $\theta^{(t)}$ adalah nilai-nilai tersimulasi dan T' adalah ukuran sampel hasil simulasi yang digunakan untuk pendugaan.

Untuk menghitung *standard error* dari penduga ini, dilakukan dengan membagi lagi urutan nilai-nilai simulasi $\theta_j^{(1)}, \dots, \theta_j^{(T')}$ menjadi b kelompok yang berukuran v , sehingga $T' = b \cdot v$. Untuk setiap kelompok dihitung rata-rata sampel, misal rata-rata kelompok sampel adalah $\bar{\theta}_j^{(1)}, \dots, \bar{\theta}_j^{(b)}$. Misalkan pula bahwa, ukuran sampel v telah dipilih cukup besar sehingga autokorelasi (lag 1) pada rangkaian *batch means* kecil, katakan di bawah 0.1, maka estimasi *standard error* $\bar{\theta}_j$ dapat diduga dengan standard deviasi dari *batch means* dibagi dengan akar kuadrat dari jumlah kelompok, yaitu:

$$S_{\bar{\theta}_j}^B = \sqrt{\frac{\sum_{l=1}^b (\bar{\theta}_j^{(l)} - \bar{\theta}_j)^2}{(b-1)b}} \quad (8)$$

Standard error ini sangat berguna untuk menentukan ketelitian dari rata-rata distribusi target yang dihitung pada simulasi yang dijalankan. Pada kejadian tersebut, jika *standard error* terlalu besar, maka algoritma MCMC sebaiknya dijalankan kembali menggunakan jumlah iterasi yang lebih besar^[4].

3. Metode Penelitian

a. Waktu dan Lokasi Penelitian

Penelitian ini dilakukan di Laboratorium Statistika, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Diponegoro dari Bulan Juni sampai dengan Oktober 2012.

b. Variabel penelitian

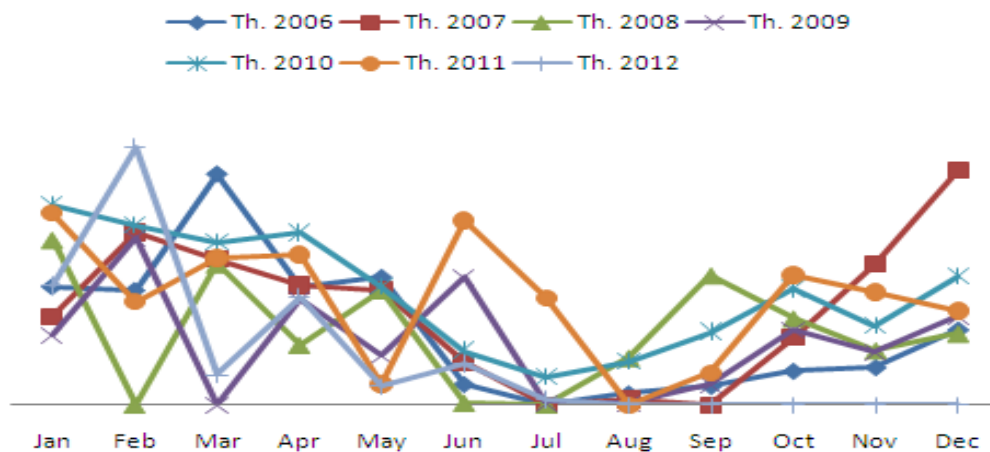
Variabel-variabel yang diamati dalam penelitian ini adalah curah hujan maksimum di Kabupaten Grobogan dari tahun 2006 sampai dengan bulan Juni 2012 yang diperoleh dari Dinas Pertanian Tanaman Pangan dan Holtikultura Kabupaten Grobogan.

c. Metode Analisis Data

- i. Eksplorasi data, yaitu mengidentifikasi pola sebaran curah hujan maksimum di Kabupaten Grobogan dengan memanfaatkan statistik lima serangkai.
- ii. Mengidentifikasi pola distribusi curah hujan melalui plot grafik dan pengujian secara statistika menggunakan uji Kolmogorov-Smirnov.
- iii. Menduga nilai parameter dari distribusi yang terpilih menggunakan metode Bayes dan MCMC dengan melalui tahapan:
 1. Mencari distribusi prior untuk parameter distribusi terpilih dengan menggunakan metode Jeffrey.
 2. Memilih distribusi proposal berupa *random walk*.
 3. Menentukan peluang penerimaan
 4. Menetapkan nilai awal
 5. Membangkitkan data
 6. Melakukan analisa konvergensi
 7. Menentukan nilai dugaan parameter
 8. Menentukan standar error nilai dugaan

4. Hasil Dan Pembahasan

4.1. Analisa Deskriptif



Gambar 1. Pola Curah Hujan dari tahun 2006 sampai dengan tahun 2012

Curah hujan maksimum di Kabupaten Grobogan dari tahun 2006 sampai dengan bulan Juli 2012 sangat bervariasi, tetapi dari tahun ke tahun memiliki pola yang cenderung tidak berubah. Curah hujan maksimum tertinggi berkisar pada bulan Desember, Januari, Februari dan Maret sedangkan curah hujan maksimum terendah biasanya berada di bulan Juni, Juli maupun Agustus.

Selama kurun waktu tujuh tahun rata-rata curah hujan di Kabupaten Grobogan mencapai 46.418 mm dengan simpangan baku sebesar 34.59 mm. Curah hujan maksimum tertinggi dicapai pada angka 138 mm yang terjadi pada bulan Februari tahun 2012 sedangkan curah hujan maksimum terendah dicapai pada angka 0 mm. Curah hujan 0 mm ini biasanya terjadi pada bulan Juni, Juli dan Agustus disepanjang tahun. Nilai kemencengan data (*skewness*) mencapai 0.4319 dan nilai kemenjuluran data (*kurtosis*) – 0.4895. Ini merupakan nilai-nilai yang termasuk rendah sehingga ada kemungkinan distribusi data curah hujan maksimum adalah distribusi yang simetri. Tabel statistik deskriptif dapat dilihat di Tabel 1. Dalam analisis frekuensi curah hujan, bentuk distribusi penting sekali untuk diketahui. Hal ini berguna untuk menentukan peluang terjadinya hujan maksimum.

Tabel 1. Statistika Deskriptif Curah Hujan Maksimum di Kabupaten Grobogan

Statistic	Nilai
Range	138.00000
Mean	46.41800
Variance	1196.50000
Std. Deviation	34.59000
Coef. of Variation	0.74518
Std. Error	3.89170
Skewness	0.43195
Excess Kurtosis	-0.48950

4.2. Identifikasi Jenis Distribusi

Distribusi statistik yang biasanya dipertimbangkan sebagai distribusi dari curah hujan maksimum adalah distribusi normal, Gumbel Maksimum, Gumbel Minimum, Log-Normal dan Log Pearson Type III^[8]. Dengan menggunakan uji Kolmogorov-Smirnov pada taraf nyata 5% diketahui bahwa curah hujan maksimum dari tahun 2006 sampai dengan 2012 di Kabupaten Grobogan mengikuti distribusi normal dengan selisih jarak antara distribusi kumulatif empiris dan teoritis sebesar 0.089. Dengan menggunakan uji yang sama, sebenarnya curah hujan maksimum ini juga dapat dikatakan mengikuti distribusi Gumbel Maksimum, Gumbel Minimum dan Log-Normal, namun karena jarak antara distribusi kumulatif empiris dan teoritis lebih besar dari 0.089 maka yang dipilih sebagai distribusi bagi curah hujan maksimum di Kabupaten Grobogan adalah normal. Dari Tabel 2 diketahui bahwa jarak antara distribusi kumulatif empiris dan teoritis dari distribusi Gumbel Maksimum sebesar 0.12039 merupakan jarak terkecil kedua setelah distribusi normal. Sedangkan Log-Normal menghasilkan jarak antara distribusi kumulatif empiris dan teoritis yang terbesar, yaitu 0.13893.

Tabel 2. Nilai Statistik Hitung dan P-Value uji Komogorov-Smirnov

Jenis Distribusi	Statistic Hitung	P-Value	Peringkat
Normal	0.08981	0.51799	1
Gumbel Maksimum	0.12039	0.18655	2
Gumbel Minimum	0.13743	0.09186	3
Log-Normal	0.13893	0.08592	4
Log Pearson Type III	Tidak Cocok	-	5

4.3. Pendugaan Parameter Distribusi

Pada bagian sebelumnya telah diketahui bahwa distribusi yang terpilih adalah distribusi normal. Menurut Casella dan Berger (2002) distribusi normal memiliki dua buah parameter, yaitu nilai tengah (μ) dan ragam (σ^2). Nilai parameter-parameter ini biasanya tidak diketahui, oleh karena itu nilai-nilai tersebut akan diduga berdasarkan informasi prior dan informasi dari sampel^[3].

Pada penelitian ini metode *Bayesian Markov Chain Monte Carlo* dipilih sebagai metode pendugaan parameter dan dalam hal ini parameter ragam dianggap diketahui. Oleh karena itu hanya parameter nilai tengah saja yang perlu diduga. Dalam kerangka Bayesian pendugaan parameter didasarkan pada distribusi posterior yang nilai-nilainya sebanding dengan hasil kali antara distribusi prior dan likelihoodnya. Likelihood data adalah hasil kali semua distribusi peluang dari setiap data. Oleh karena distribusi peluangnya adalah normal maka untuk setiap nilai sampel distribusi peluangnya adalah

$$f(x_i | \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i - \mu}{\sigma}\right)^2} \quad \text{untuk } i = 1, 2, \dots, n$$

dan fungsi likelihoodnya adalah

$$L(\mu | x_1, x_2, \dots, x_n; \sigma^2) = \prod_{i=1}^n f(x_i | \mu, \sigma^2) = \frac{1}{\sigma^n (2\pi)^{n/2}} e^{-\frac{1}{2} \sum_{i=1}^n \left(\frac{x_i - \mu}{\sigma}\right)^2}$$

Dalam penelitian ini distribusi prior bagi nilai tengahnya ditentukan dengan menggunakan metode Jeffrey. Prior Jeffrey adalah salah satu jenis prior *uninformative* dimana dipilih apabila informasi awal mengenai parameter distribusi sangat kurang. Proses untuk menentukan prior Jeffrey adalah sebagai berikut:

$$f(x|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$$\ln f(x|\mu, \sigma^2) = -\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 + \ln\left(\frac{1}{\sigma\sqrt{2\pi}}\right)$$

$$\frac{\partial \ln f(x|\mu, \sigma^2)}{\partial \mu} = -\frac{1}{2} \cdot 2 \left(-\frac{1}{\sigma}\right) \cdot \left(\frac{x-\mu}{\sigma}\right) = \frac{x}{\sigma^2} - \frac{\mu}{\sigma^2}$$

$$\frac{\partial^2 \ln f(x|\mu, \sigma^2)}{\partial \mu^2} = -\frac{1}{\sigma^2}$$

Prior Jeffrey diperoleh dengan mengambil akar kuadrat dari informasi Fisher, yaitu

$$I(x) = -E\left(\frac{\partial^2 \ln f(x|\mu, \sigma^2)}{\partial \mu^2}\right)$$

Sehingga distribusi prior bagi nilai tengah curah hujan di Kabupaten Grobogan adalah sebanding dengan

$$\propto \sqrt{-E\left(\frac{1}{\sigma^2}\right)} = \sqrt{\frac{1}{\sigma^2}} = \frac{1}{\sigma}$$

Nilai simpangan baku diduga dengan menggunakan informasi dari sampel, yaitu sebesar 34.589.

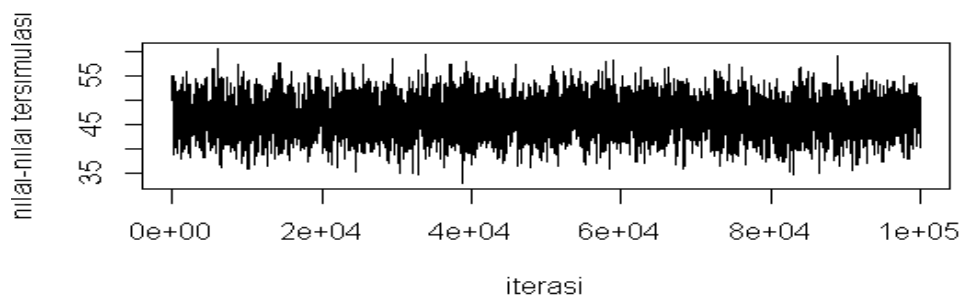
Selanjutnya akan ditentukan distribusi posterior bagi nilai tengah curah hujan maksimum di Kabupaten Grobogan yang nilainya sebanding dengan hasil kali antara distribusi prior dan fungsi likelihood, yaitu

$$L(\mu|x_1, x_2, \dots, x_n; \sigma^2) \propto \frac{1}{\sigma^n (2\pi)^{n/2}} e^{-\frac{1}{2}\sum_{i=1}^n \left(\frac{x_i - \mu}{\sigma}\right)^2} \cdot \frac{1}{\sigma}$$

Pendugaan parameter nilai tengah didasarkan pada distribusi posterior di atas. Algoritma Metropolis-Hastings digunakan untuk membangkitkan nilai-nilai stokastik yang dianggap berasal dari distribusi posterior tersebut. Distribusi proposal yang digunakan adalah Normal dengan simpangan bakunya 200. Nilai tengah distribusi proposal ini bersifat *random walk*, yaitu bergantung pada nilai simulasi sebelumnya yang terpilih sebagai anggota sampel. Distribusi proposal ini digunakan untuk membangkitkan kandidat sampel dari posterior yang kemudian akan dievaluasi diterima maupun tidaknya. Jika nilai awal bagi μ adalah μ_0 dan nilai μ yang dibangkitkan adalah μ' maka peluang menerima μ' sebagai anggota sampel posterior adalah

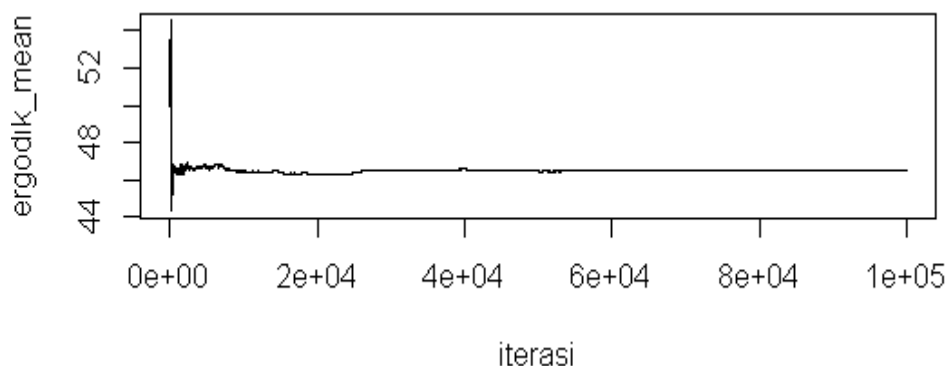
$$p = \frac{\frac{1}{\sigma^n (2\pi)^{n/2}} e^{-\frac{1}{2}\sum_{i=1}^n \left(\frac{x_i - \mu'}{\sigma}\right)^2} \cdot \frac{1}{\sigma}}{\frac{1}{\sigma^n (2\pi)^{n/2}} e^{-\frac{1}{2}\sum_{i=1}^n \left(\frac{x_i - \mu_0}{\sigma}\right)^2} \cdot \frac{1}{\sigma}} = \frac{e^{-\frac{1}{2}\sum_{i=1}^n \left(\frac{x_i - \mu'}{\sigma}\right)^2}}{e^{-\frac{1}{2}\sum_{i=1}^n \left(\frac{x_i - \mu_0}{\sigma}\right)^2}}$$

Proses membangkitkan sampel posterior akan terus dilakukan sampai konvergensi dari iterasi tercapai. Dalam penelitian ini digunakan nilai awal bagi μ adalah 50. Pemantauan hasil dari konvergensi iterasi dapat dilihat di Gambar 2 sampai dengan Gambar 5.



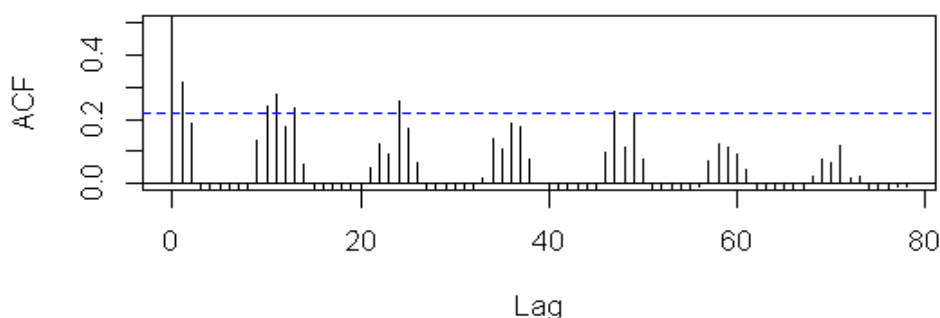
Gambar 2. Trace Plot Hasil Simulasi

Trace plot merupakan plot antara banyaknya iterasi yang dijalankan dengan sampel yang dibangkitkan. Pada Gambar 2 tersebut terlihat bahwa *trace plot* tidak membentuk suatu pola atau *trend*. Hal ini menunjukkan bahwa *burn-in period* sudah selesai yang berarti sampel yang dibangkitkan telah berada dalam daerah distribusi target. Oleh karena itu, dapat dikatakan bahwa algoritma mencapai konvergensi.



Gambar 3. Ergodic Mean Plot

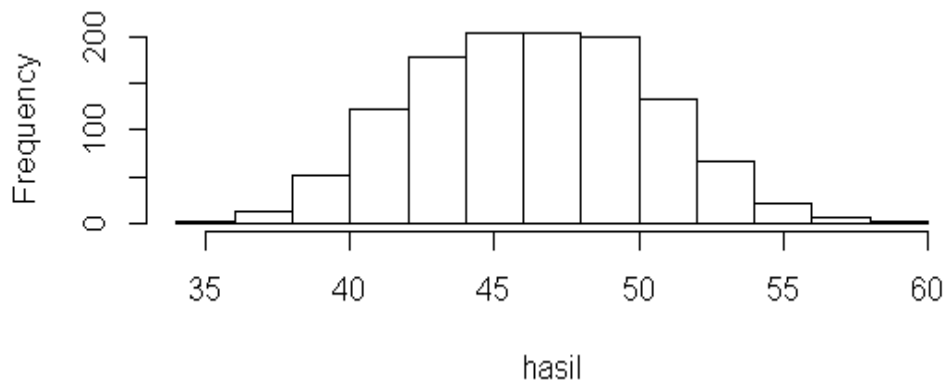
Plot *ergodic mean* merupakan plot antara nilai mean sampai dengan *current iteration* dengan iterasinya. Pada Gambar 3 terlihat bahwa grafik tidak membentuk pola atau *trend* tertentu dan telah stabil pada iterasi ke-40000 (*burn-in period*). Kestabilan dari plot *ergodic mean* merupakan indikasi algoritma telah mencapai konvergensi dan sampel yang memenuhi distribusi target untuk dibangkitkan mulai dari iterasi ke-40001 sampai dengan itersi ke 100000.



Gambar 4. Autocorrelation Function Plot

Plot fungsi autokorelasi (*ACF Plot*) digunakan untuk mengidentifikasi kuat lemahnya korelasi yang terjadi diantara nilai-nilai sampel yang dibangkitkan. Pada Gambar 4 terlihat

bahwa nilai autokorelasi pada lag pertama mendekati satu dan nilai-nilai selanjutnya semakin mendekati 0. Hal ini menunjukkan bahwa korelasi diantara nilai-nilai sampel yang dibangkitkan lemah, dengan kata lain nilai-nilai tersebut telah berada di daerah distribusi posterior.



Gambar 5. Histogram Nilai-Nilai Posterior

Tidak semua nilai tersimulasi setelah proses *burn- in period* digunakan untuk pendugaan parameter. Dalam penelitian ini digunakan *thinning interval* sebesar 50 yang berarti nilai-nilai yang dipilih adalah nilai yang tersimulasi pada iterasi ke-40001, 40051, 40101 dan seterusnya. Dengan adanya *thinning interval* diharapkan nilai-nilai yang digunakan untuk pendugaan parameter bersifat bebas antara satu dengan yang lainnya. Gambar 5 merupakan histogram nilai-nilai yang dianggap berasal dari distribusi posterior. Data tersebut akan dijadikan dasar untuk menduga parameter nilai tengah dari distribusi curah hujan maksimum yang hasilnya adalah

$$\hat{\mu} = \frac{\sum_{t=1}^{T'} \mu_t}{T'} = 46.269 \text{ mm}$$

dengan simpangan baku sebesar 4.005 mm.

5. Kesimpulan

Berdasarkan pembahasan, maka dapat diambil kesimpulan sebagai berikut:

1. Curah hujan maksimum di Kabupaten Grobogan dari tahun 2006 sampai dengan bulan Juli 2012 sangat bervariasi, tetapi dari tahun ke tahun memiliki pola yang cenderung tidak berubah. Curah hujan maksimum tertinggi berkisar pada bulan Desember, Januari, Februari dan Maret sedangkan curah hujan maksimum terendah biasanya berada di bulan Juni, Juli maupun Agustus.
2. Dengan menggunakan uji Kolmogorov-Smirnov pada taraf nyata 5% diketahui bahwa curah hujan maksimum dari tahun 2006 sampai dengan 2012 di Kabupaten Grobogan mengikuti distribusi normal dengan selisih jarak antara distribusi kumulatif empiris dan teoritis sebesar 0.089. Jarak ini menghasilkan nilai signifikansi sebesar 0.51799.
3. Dengan menggunakan metode Bayesian Markov Chain Monte Carlo diperoleh bahwa nilai dugaan untuk parameter nilai tengah distribusi normal adalah 46.269 mm dengan simpangan baku sebesar 4.005 mm.

DAFTAR PUSTAKA

1. Bain, L. J., and Engelhard, M., *Introduction to Probability and Mathematical Statistics*, Second Ed., California. Duxbury Press, 1992, p:292-301.
2. Box, G.E.P. and George, C.T., *Bayesian Inference in Statistical Analysis*, Addison-Wesley Publishing Company, Inc., Canada, 1973, p:1-20.
3. Casella, G. and Berger, R.L., *Statistical Inference*, Second Ed., Thomson Learning, Duxbury, 2002, p: 435-440.
4. Johnson, V. E. and Albert, J. H., *Ordinal Data Modelling*, Springer-Verlag Inc., New York, 1998.
5. Ntzoufras, I., *Bayesian Modeling Using Winbugs*, John Wiley & Sons, New Jersey, 2009.
6. Roldan, J. and David, A.W., Stochastic Daily Models: A Comparison of Occurrence Processes, *Water Resources Research*, 1982, Vol.18, No.5: 1451-1459.
7. Soewarno, *Hidrologi Operasional*, Jilid kesatu, Citra Aditya Bakti, Bandung, 2000.
8. Suhaila, J. and Abdul, A.J., Fitting the Statistical Distribution for Daily Rainfall in Peninsular Malaysia Based on AIC Criterion, *Journal of Applied Sciences Research*, 2008, Vol 4, No.12: 1846-1857.
9. Walpole, R. E., *Pengantar Statistika*, Edisi Keenam, Gramedia Pustaka Prima, Jakarta, 1995.